ABC on LCD via PC port LPT

Liquid Crystal Displays are used everywhere these days. But how do they work? We'll show you how to drive an LCD from your PC keyboard. Then we show how to use the LCD to display temperature. It's a great project for beginners, especially if you want to understand more about your computer.

By Peter Crowcroft

You'll find LCDs, or liquid crystal displays, in a huge variety of appliances, consumer electronics, and so on. Usually driven by a microcontroller, they've become very popular over recent years for information transfer and instructions and (usually!) make complicated equipment easier to operate.

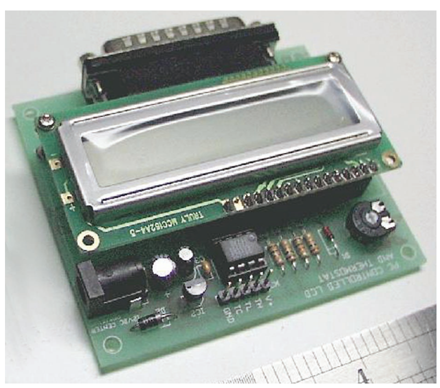
But how do you get the message you want onto the LCD screen? It's not difficult, as this simple project shows.

Circuit description

The circuit (Fig.1) is quite straightforward. In fact, most of the "work" is undertaken by the large integrated circuit hidden under the black blob on the rear of the LCD module PC board.

All we have to do is provide that module with the appropriate ASCII codes (and these come from your computer's LPT1, or parallel, port).

The project could have been as simple as that but we've gone one step further and added a temperature control chip, the DS1620, which we can use to make a simple but effective thermostat/thermometer in conjunction with the LCD module.



The remainder of the circuitry is mainly power supply – and even that is very simple.

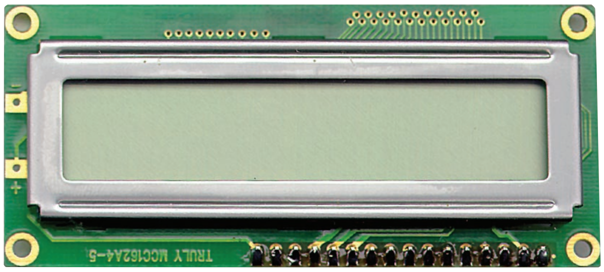
Power is supplied from a 12V DC plugpack, with diode D2 protecting against reverse polarity on the DC input jack (centre pin is positive). Due to the low current requirements of the kit, only a small (TO-92 pack) 5V regulator is required.

Resistors R2 and R3 hold the DS1620 CLK and RST inputs low when the kit is disconnected from the PC. This allows the chip to work in "stand-alone" mode.

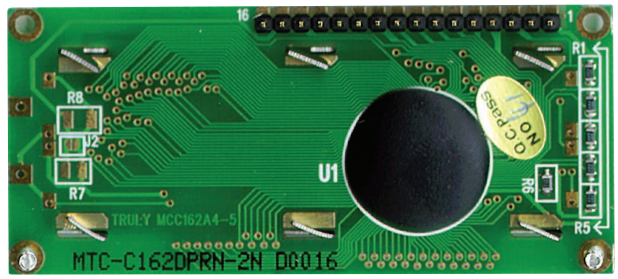
Resistor R1 and diode D1 convert the separate data input and output lines from the PC to the single bidirectional data line on the DS1620. When writing to the chip, R1 pulls DQ high when AUTO is high. DQ is pulled low via D1 when AUTO is low. When reading from the DS1620, AUTO must be set high first. This high is blocked by D1 so that the DQ output can drive ACK.

The liquid crystal display

LCDs come in many shapes and



Front view of the pre-built Liquid Crystal Display (LCD) module used in this project. It has 2 lines of 16 characters. All these pics are shown same size.



The same module viewed from the rear. Under that big black blob is a chip which does all of the work in driving the display. All you have to do is talk to it!

sizes but amongst the most common is the type we have used in this project, a 16 character x 2 line display.

It requires only 11 connections – eight bits for data (which can be reduced to four if necessary) and three control lines (we have only used two here).

This particular display, an MC-162-2, runs from a 5V DC supply and only needs about 1mA. The display contrast can be varied by changing the voltage into pin 3 of the display, usually with a trimpot.

The display requires eight bits of data, a register select line (RS) and a strobe line (E), which are supplied from the PC printer port. A third input, read/write (R/W), is normally used to read or write data to and from the LCD. In this kit the R/W line is tied low so only "writes" to the LCD are possible (more on this later).

Here's what you get in the kit of parts. On top is the LCD module, pre-assembled, with the main PC board underneath. The software is on the 3.5-in floppy while all the components are separately packed in plastic. Inset at top is the kit as supplied.

The eight bits of data are supplied from the printer port data lines and two printer port control lines are used for RS ('auto') and E ('strobe').

Basically the LCD has two registers, a data register and a control register. Data is written into the control register when RS is low and into the data register when RS is high. Data is latched into the LCD register on the falling edge of 'Enable'.

The sequence for writing data to the LCD is:

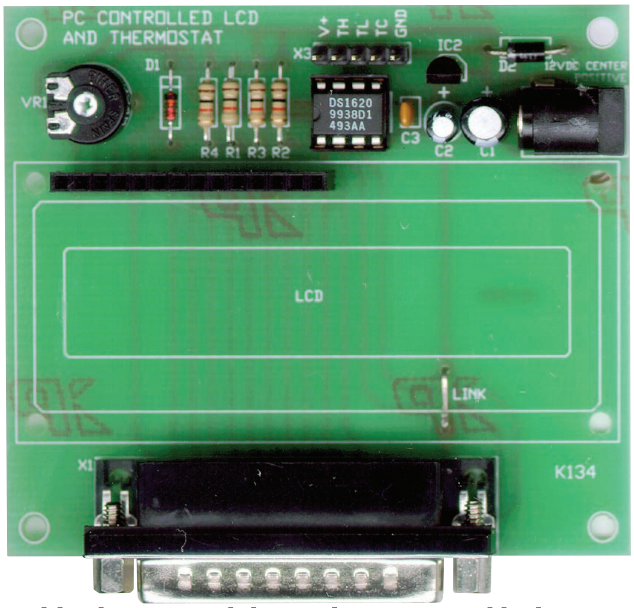
1. To begin, E is low
2. Select the register to write to by setting RS high (data) or low (control)
3. Write the eight bits of data to the LCD
4. Set the Enable signal high then low again.

There are certain minimum timing requirements that must be followed when writing to the LCD, such as data setup times and Enable signal pulse width. These are in the order of tens and hundreds of nanoseconds.

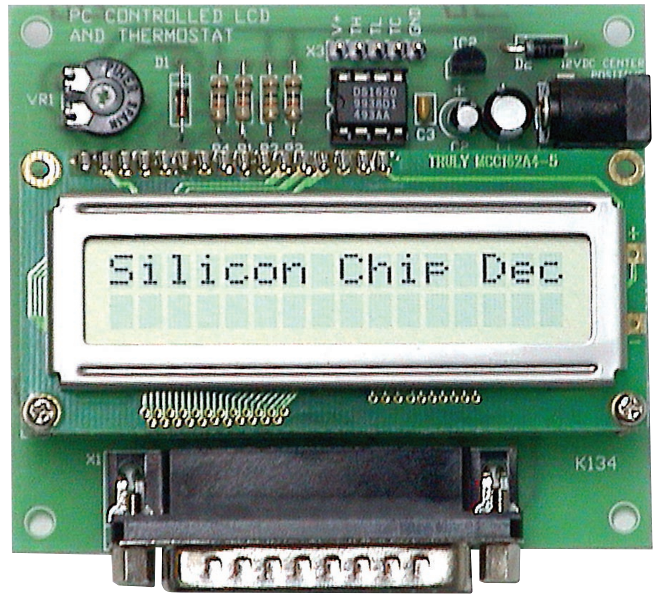
Extra timing delays are also introduced by the capacitance of the cable used to connect to the PC printer port. Delay routines may be needed when using fast PCs to meet these timing requirements. The software supplied includes these delay routines.

LCD displays have a 'busy' flag that is set while it is executing a control command but in our case this flag is not accessible because the R/W line has been tied low (write).

This is not such a problem because all commands have a maximum execution time. We simply wait for this



Unlike the LCD module, you have to assemble this PC board yourself. But as you can see, there's not much to it. The most difficult bit is the D-25 socket at the bottom.



And here it is with the LCD module plugged in and running. Perhaps you're wondering "how did they get a message on the screen when nothing is plugged in...?"

time to pass before accessing the LCD again, eliminating the need to check the flag and thereby saving on an I/O line!

For example, the 'Clear Display' command has an execution time of approx. 1.6mS. After sending this command to the LCD we simply wait 2mS before continuing on. This ensures that the command has finished.

Characters to be displayed are written into the LCD's 'data' memory (RAM).

The amount of RAM available depends on the type of LCD. The LCD used in this kit has 80 bytes of RAM. An internal address counter holds the address of the next byte to write to.

These 80 bytes are broken up into two blocks of 40 bytes. The address range of the first block is from 00h to 27h and the second block from 40h to 67h.

After power up and initialization address 00h is the first character of the top line and address 40h is the first character of the bottom line. The address counter is set to address 00h and is automatically incremented after each byte is written.

If we now start writing data to the LCD it will be stored from address 00h on but only the first 16 characters will be visible. To display the rest of the characters we need to 'scroll' the display.

Scrolling simply means changing the start address of each line. If we scroll left one position address 01h

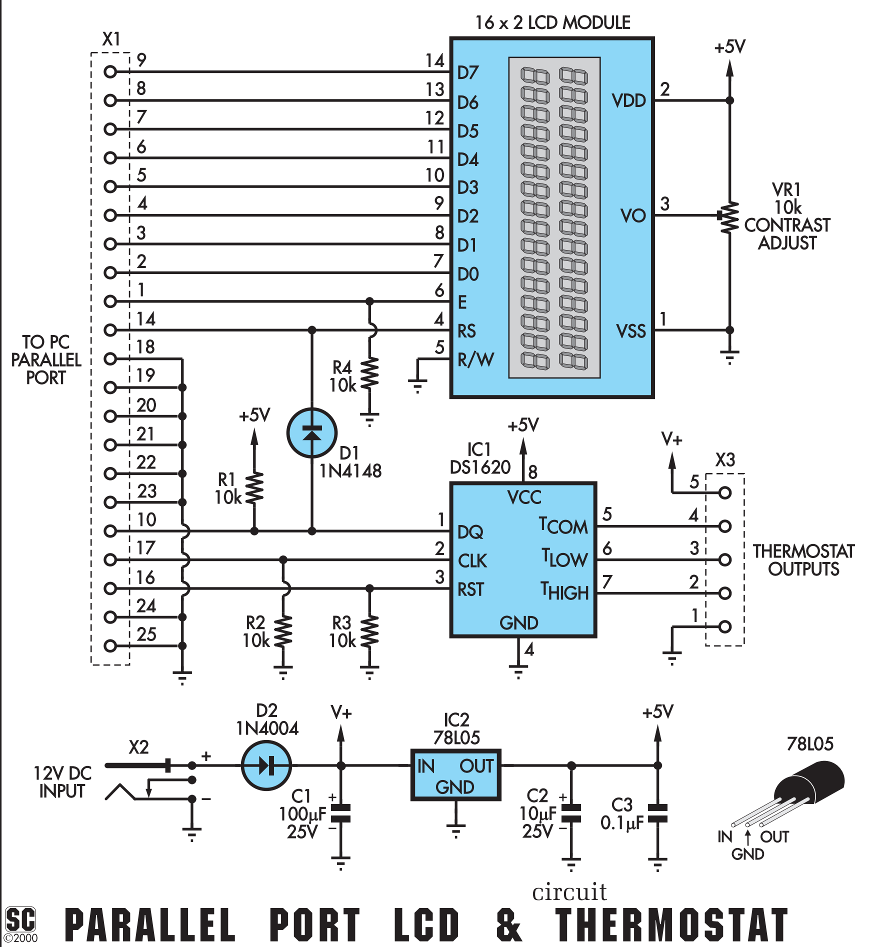


Fig.1: the circuit is quite simple – just the 5V regulated power supply to drive the LCD module and a DS1620 thermometer chip which displays its readings on the LCD screen.

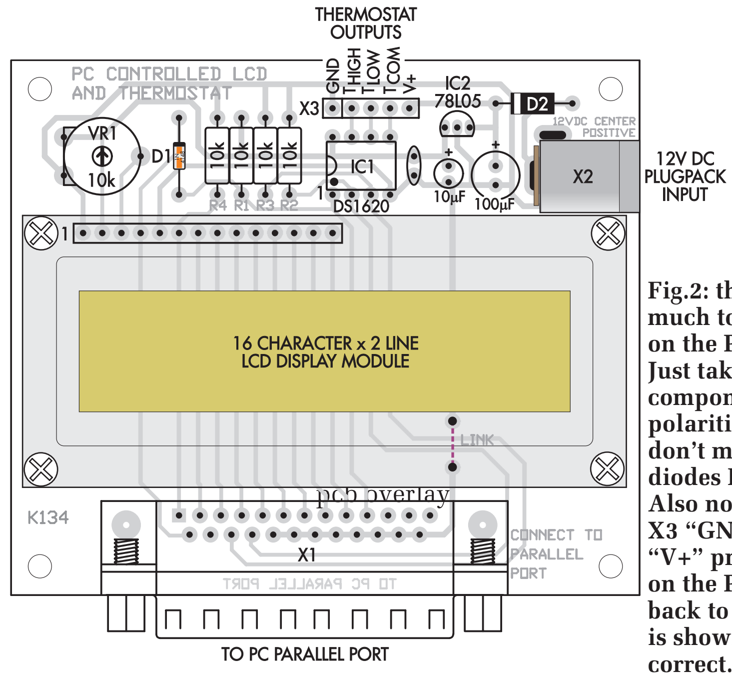


Fig.2: there's not much to assemble on the PC board. Just take care with component polarities – and don't mix up diodes D1 and D2! Also note that the X3 "GND" and "V+" printed labels on the PC board are back to front: what is shown here is correct.

becomes the first character of the top line and address 41h the first character of the bottom line.

Scrolling right does the opposite – addresses 27h and 67h become the first characters of the top and bottom lines.

As you can see, each block of addresses 'wrap' around in a circular manner. The next address after 27h is 00h; the next address after 67h is 40h. If we scroll left 40 times we will bring address 00h back into view as the last character on the top line.

The address ranges of each block are not continuous. There is a gap of 24 bytes between the end of the first block (27h) and the start of the second (40h).

So if we keep writing characters to the display the first 40 will be stored in addresses 00h to 27h, the next 24 will be 'lost' and the next 40 will be stored at addresses 40h to 67h.

The internal address counter is directly accessible so we can set the next address to write to. This is how the characters are written to the second line of the LCD. For example, we can send 5 characters to the first line, set the address counter to 40h and then start writing characters to the second line.

The DS1620 chip

The DS1620 is a Digital Thermometer/Thermostat IC from Dallas Semiconductor. It measures temperatures from -55° to +125°C (-67° to +257°F) in 0.5°C increments and has three "alarm" outputs. It requires no calibration or external components.

(If you want to know more about this chip you can download the data sheet from *www.dalsemi.com*).

The three alarm outputs, designated T_{HIGH}, T_{LOW} and T_{COM} function as follows:

- T_{HIGH} goes high when the measured temperature is greater than or equal to a user-defined upper limit (TH).
- T_{LOW} goes high when the measured temperature is less than or equal to a user-defined lower limit (TL).
- T_{COM} goes high when the temperature exceeds the upper limit and stays high until the temperature falls below the lower limit.

These three outputs can be used to directly control heating and cooling appliances via suitable drive circuitry (eg. relays and optocouplers).

Data is read from and written to the DS1620 via a 3-wire serial interface (DQ, CLK and RST). The user-defined upper and lower trip points are stored in the IC in non-volatile memory. This means that the trip points are 'remembered' even when the power is removed.

The DS1620 can be used in two modes – Three-Wire mode for interface to a computer or Stand-Alone mode.

This last mode does not require a computer interface. With RST and CLK low, the chip will continuously monitor the temperature and set the alarm outputs accordingly.

This means that once you have programmed the upper and lower trip points the kit can be disconnected from the PC and you can use the alarm outputs for monitoring or control functions. You can even remove the chip from its socket and put it in your own temperature controller, using the kit simply as a 'DS1620 programmer'.

Temperature registers in the DS1620 are nine bits long and can be positive or negative values. Each reading has a resolution of 0.5° C. For positive temperatures, the most significant bit (bit 8) = 0 and bits 1-7 hold the integer value. Bit 0, if set, adds 0.5°.

For negative temperatures, bit 8 = 1 and bits 7-0 represent the two's complement of the temperature. To get the actual value, invert bits 7-0 then add 1.

The DS1620 is controlled via a set of 8-bit instructions. Depending on the instruction an 8 or 9 bit data value is either read from or written to the chip. All instructions and data are transferred serially starting with the least significant bit D0.

To read or write a value to the DS1620:

1. To begin, RST is low and CLK high.
2. RST high
3. CLK low
4. Set DQ equal to Bit 0 of instruction
5. CLK high
6. Repeat steps 3, 4 and 5 for remaining instruction bits

To write to the chip:

7. Repeat steps 3, 4 and 5 for Bits 0-7 or 0-8 of the data to be written
8. RST low for at least 5ms

To read from the chip do Steps 1 through 6 above then:

7. Set the AUTO signal high so that data can be read from the DS1620
8. CLK low. DQ now outputs the data to be read (LSB first)
9. Read and store DQ
10. CLK high
11. Repeat steps 8, 9 and 10 for the remaining bits of data to be read
12. RST low

As with the LCD, certain minimum timing requirements must be observed between each of these steps. In particular, RST must remain low for at least 5ms after writing data to the DS1620 (step 8). This gives the chip's internal EEPROM time to store the data. Also, the DS1620 needs 1 second to execute a 'start-convert' in-

Here's how the LCD module mounts to the main PC board. The back of the module has a 16-way header pin set which mates with a 14-way socket on the board.

struction. So, after starting a temperature conversion, you must wait 1 second before reading the result.

The PC parallel port

There are three common address ranges used for parallel port interfaces, as follows:

Table 1 – Parallel Port Addresses

3BCh - 3BFh	Parallel interface on monochrome or TTL video card
378h - 37Fh	Parallel interface 1 (LPT1)
278h - 27Fh	Parallel interface 2 (LPT2)

The standard PC parallel port consists of three registers, each referred to by their functional name. These registers occupy the first three addresses in the range. The first address in the range is referred to as the ***base address***.

Table 2 – Base Registers

Data register	Base address	Read/Write
Status register	Base address + 1	Read only
Control register	Base address + 2	Read/Write

For example, for parallel interface 1, the data register would be at address 378h, status register at address 379h and control register at address 37Ah.

Note that the data and control registers can be written to and read from. This does not mean that they are bidirectional. It means that the outputs of these registers can be read back to check their status.

During the system boot-up sequence, the BIOS tests for the presence of parallel ports in order, according to Table 1. The first port found becomes LPT1, the second LPT2, and so on. The BIOS stores the base address of each port found in a table in the BIOS variable segment of memory, as follows:

Table 3 – LPT Base Addresses

ADDRESS	CONTENTS
0040:0008H	Base address of LPT1
0040:000AH	Base address of LPT2
0040:000CH	Base address of LPT3
0040:000EH	Base address of LPT4

(If address=0 then there is no port for that LPT number.)

Every signal line on the port, whether it is used for read/input or write/output, is allocated one particular bit at one of the three addresses Base, Base+1 or Base+2. The logic state of the bit indicates the state of the wire (0V or 5V.) The following table shows each printer port signal used by the kit and the register, address and bit position associated with it. The last three signal names are preceded by a minus sign. This means that the signal is active low. Writing a high to the bit causes the signal to go low.

Table 4 – Signal Line Attributes

Signal	Reg.	Addr.	Bit	Direction
D0 D7	Data	Base	0 - 7	Output
ACK	Status	Base+1	6	Input
-STROBE	Control	Base+2	0	Output
-AUTO	Control	Base+2	1	Output
INIT	Control	Base+2	2	Output
-SLCTIN	Control	Base+2	3	Output

Construction

There aren't very many components to place on the PC board so chances of errors aren't high – another feature which makes this a great project for beginners. On the negative side, though, your soldering skills will be tested!

As with all projects, before placing or soldering components, check the PC board for obvious defects – bridges or shorts between tracks which shouldn't be there, broken tracks, etc.

Start with the lowest profile components – the resistors and the wire link (use a resistor offcut for this). The two diodes are next but take care that you not only get them in the right places but in the right polarity. The power diode is the larger black diode and it is the one which goes in the top right corner of the board.

Place the capacitors next, again taking care with type and polarity. If you haven't used electrolytic capacitors before, the negative lead is marked by a row of "–" symbols down the side of the capacitor adjacent to that lead.

Now solder in the trimpot, voltage regulator and DC socket. The trimpot and DC socket can only go one way but the voltage regulator could be mounted back-to-front if you're not careful: it must go in the same way as indicated on the board.

Mount the 5-way header pin set and the IC socket – again, make sure that the IC socket is oriented the same way as indicated on the board.

Parts List – LCD Interface

1 PC Board K134, x mm
1 Liquid Crystal Display module, 16 x 2 characters

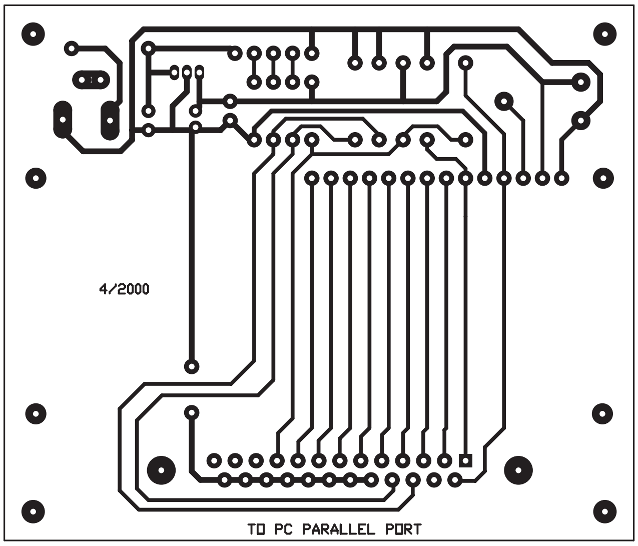
Semiconductors
1 1N4004 (D2)
1 1N4148 (D1)
1 78L05 +5V regulator, TO-92 package (IC2)
1 DS1620 digital thermometer and thermostat (IC1)

Capacitors
1 100uF 25V electrolytic (C1)
1 10uF 25V electrolytic (C2)
1 100nF monobloc (C3)

Resistors (0.25W carbon)
4 10KΩ (R1-4)
1 10KΩ trimpot (VR1)

Miscellaneous
1 2.5mm DC jack, PCB mounting (X2)
1 D25 male connector, right-angle PCB mounting (X1)
1 5-pin SIL header (X3)
1 8-pin IC socket for IC1
1 14-pin SIL socket for LCD
1 14-pin SIL header for LCD
2 screws, 2.6mm x 18mm long
6 nuts, 2.6mm
1 software floppy disk

A complete kit of parts, including software, is available from all Jaycar Electronics stores (Cat KD-6082) $69.95



Use this same-size PC board artwork to check the board supplied in your kit – just in case.

When soldering the D connector, IC socket and header pins, you will need to be particularly careful not to form solder bridges between the pins. A fine-tipped, clean iron is essential – and if you don't want to strain your eyes, some form of magnifying lens is very handy (especially for us older dudes whose close-up eyesight ain't what it used to be!).

You should now have only a couple of components left: a header pin set and matching socket and a 25-pin D connector. The D connector might cause you a little bit of angst because it has to be very carefully aligned to go into its holes. You cannot put it in at an angle – the pins at the opposite end will miss their holes. But it does go in!

The last component to mount on the main PC board is the 14-way header pin socket (ie, the larger of the two header components).

The 16-way header pin set has to be soldered to the display module containing the display. This needs to be exactly vertical and is soldered on the same side as the LCD (ie, the header pins emerge from the underside of the board. At the risk of boring you, we must repeat – these pins are very close together. Watch out for solder bridges!

Just in case you're wondering, no, we haven't made a mistake with the number of pins in the header pin and sockets. The pin set has 16 pins, the socket 14 pins. 2 pins are unused in this project and simply hang off the end of the socket when later assembled.

Don't assemble it yet – nor for that matter, insert the DS1620 IC. We're going to check the board first.

But even before that, give your board the once-over to make absolutely sure your component placement is correct and that you have neither missed any pins while soldering nor soldered any bridges. Obviously, if you find any mistakes, correct them first.

It's best to take your time and not make any mistakes in the first place. But if you do have to unsolder components – especially things like PC header pins or IC sockets – you'll find it almost impossible without using either solder wick (which soaks up the solder, freeing the pins) or a solder sucker (which does the same thing by suction).

Power

This project is designed to run from a 12V DC supply which is then regulated to 5V by the 78LS05. An ideal supply would be a 12V plugpack – because the circuit draws so little power, just about any 12V DC plugpack you find should be OK.

BUT! (Don't you just hate those buts?) The DC socket on the board is designed to suit 2.5mm plugs and, try as we might, we couldn't find a 12V DC plugpack in our junkbox with a 2.5mm plug. Plenty of 2.1mm – which don't quite fit – but no 2.5's. If you're in the same position, the obvious answer is to cut the 2.1mm plug off and solder on a 2.5mm plug.

Just one thing – make sure the centre contact is the positive contact. If it's the other way around you won't do any damage (thanks to the protective diode) but it won't work, either …

> **Note:**
> On the PC board silk-screened overlay, the two end pins of header pin set X3 are labeled incorrectly: the GND (ground) pin is the left-most pin and the V+ pin is the right-most pin with the D25 connector at the bottom. You have been warned! Incidentally, that V+ is not the regulated 5V supply but the full supply voltage from the nominal 12V plugpack – which could be as high as 16V or so.

Testing and final assembly

This is simply a check that the project powers up as it should. Without the IC or LCD module plugged in, connect your 12V DC source and measure the voltage between pins 4 and 8 of the IC socket (see the diagram to determine which pins those are). You should get a reading very close to 5V (between about 4.9 and 5.1V). If not, disconnect the power and check your PC board once again.

If you get 5V, disconnect power, wait a minute or so for the capacitors to discharge, then plug in the IC (watch its orientation!) and the LCD module.

As we mentioned before, the two pins closest to the middle of the module are not used and simply hang out in mid-air the end of the 14-pin socket.

Reconnect power and you should find a row of black squares on the display. If you don't, adjust the pot until you do. (The other thing that affects the contrast of the display is the angle at which you're viewing it. Best is straight on).

Once you're happy that it's working, remove the display module from its socket and insert screws in the two uppermost corner holes. Place a nut on each of these and tighten fully. Now place another nut on each screw and move it to about half-way along the thread.

Replace the module in its header socket, with the ends of the screws through the appropriate holes in the PC board underneath.

Rotate the two nuts so that they lie on the top side of the PC board underneath, with the two boards exactly parallel. Then insert the remaining two nuts on the ends of the screws and completely tighten.

Screw holes are provided on the top side of the display PC board and the board underneath but they are redundant as the display board is held securely in place by its header socket.

So far so good. You're now ready to fire up the PC and drive the display.

Connecting to the computer

Here's a couple of real traps for young players. And not-so-young, either, 'cos we wuz trapped!

First of all, the cable: the module is fitted with a 25-pin D "male" connector and your computer's parallel port (LPT1) will be fitted with a 25-pin D "female" connector. So all you need is a 25-pin D male to female lead, right?

Yes . . . and no!

There are two types of commonly available 25-pin D leads. One is simply a "straight through" connection (eg pin 1 to pin 1, pin 10 to pin 10, etc) and that is the type of lead you require.

The other lead is intended for serial ports (some of which also use 25-pin D connectors!) and they are NOT straight through wiring: some of the lines cross over to other pins.

Guess which lead we grabbed from our box of various computer leads? Of course, Murphy's law applied and we tried to use the serial lead. And just as "of course", it didn't work.

So make sure the lead you use is the right one.

There is a second, less obvious trap. And again, we got caught. The software is written to suit a computer with a parallel port (LPT1) at address 378H. Now 99% of computers will have their LPT1 at this address, set ex-factory. There are some computers, though, which have their LPT1 at 278H. In normal circumstances, it doesn't matter – Windows for example will work fine with either.

But this display won't. It needs 378. And guess which address our computer used? Yes, it was 278. (To be fair, the PC is an old 486/50 "workhorse" which we use for all sorts of project development and testing, leaving our main networked system for magazine production. Someone at some time had a reason to change LPT1 and it had never been changed back again!).

So if you fire up the software and it doesn't work, those are the two most obvious reasons why not. The first problem is fixed by swapping leads. The second is either a motherboard jumper change or (in more modern computers) with a BIOS change at boot-up.

Software

Two programs are supplied with the kit and are also available for downloading. Both are zipped into one archive and will need to be unzipped first.

Create a directory on your hard disk and copy the zip file from the floppy to it. Then use one of the many unzipping programs (eg WinZip, PKunzip, etc) to restore the files. Along with several "jpg" images which show how the kit is put together (some of which are reproduced here) are two .exe files which are the demonstration files for use with the display.

There are other files including "C" files written under Borland Turbo C for DOS. The .exe files (which are compiled versions of the C files) are designed to run under DOS (remember that?) but also run quite happily in a DOS box under Windows 95 or 98.

As most computers these days are Windows boxes, it's probably easier to use them this way. Simply double click on one of the appropriate .exe files and a DOS box will open, running the software. The two demo programs are:

K134LCD.EXE – this simply allows the user to enter a message to be displayed on the LCD. The message can be up to 40 characters long. The message can be stopped or scrolled left or right and the scrolling rate can be varied.

When the program is run a menu is displayed on the PC from which you choose the required function. At the same time, the square boxes on the display change to a message "PC controlled LCD demonstration".The message is only displayed on the top line of the display.

If the message is scrolling and it is less than 40 characters long then a number of spaces will be displayed before the message starts again.

You can enter a new message by selecting option 1 on the PC menu, change the scrolling direction (2 and 3), stop scrolling (4), change the scrolling speed (5) or exit the program (0). The full ASCII set available from the computer keys is echoed on the display.

Disconnecting the display from the serial port while a message is being displayed freezes the display at that point but does not lose the information - reconnection will start the message scrolling again, if that is what has been selected.

K134TEMP.EXE – a "thermostat" program that displays the current temperature as well as the upper (TH) and lower (TL) temperature 'trip' points, using the DS1620 IC mounted on the same PC board as the display components.

The value of the trip points can be changed via the PC keyboard and the user can choose between Celsius or Fahrenheit display.

The DS1620 is configured to perform continuous temperature conversions. Current temperature and trip point data is continually read from the DS1620 and compared with the previous reading. If any of the three values have changed then the display is updated.

At the same time, the outputs (via header pin set X3) will reflect the status, as previously explained.

These three pins can be used to control external equipment – eg, an alarm or a heating element if the temperature goes too low.

References

These days the internet is the place to get information. But here are some magazine references on LCD's:

"A liquid crystal display driven from a PC printer port", ***SILICON CHIP*, *March 1998***.

"A PC-controlled thermometer/thermostat", ***SILICON CHIP*, *June 1997.***

"Temperature Monitoring With a Synchronous Serial Link", ***MicroComputer Journal March/April 1995.***

Some websites you might like to visit are:

beyondlogic.org (excellent, must-visit site)

geocities.com/ResearchTriangle/1495/ee_lcd.html

iaehv.nl/users/pouweha/lcd.htm

home.nikocity.de/woe/lcd

pobox.com/~lcd_info

Or simply do a search on your favourite search engine (eg *yahoo.com*) for 'LCD parallel' and you will get hundreds of links to follow-up.

You can email the author at *peter@kitsrus.com* if you have any problems or requests.

Information on other kits in the range is available from his web page at *www.kitsrus.com* **SC**